Homework ada7

TMI M1 37-176839 Koichiro Tamura

homework1

ガウスカーネルモデル

$$q(y|\mathbf{x};\theta^{(y)}) = \sum_{j:y_j=y} \theta_j^{(y)} exp(-\frac{\|\mathbf{x} - \mathbf{x_j}\|^2}{2h^2})$$

に対して最小二乗確率的分類を実装せよ

answer

In [36]:

```
%matplotlib inline

import numpy as np
import pandas as pd
import math
import random
import matplotlib.pyplot as plt
from sklearn.utils import shuffle
from sklearn.metrics import *
from sklearn.preprocessing import OneHotEncoder
```

In [33]:

```
class GaussKernelModel(object):
    """GaussKernelModel of probability """
    def __init__(self, h=0.3, _lambda=0.1):
        # hyperparameter
        self.h = h
        self._lambda = _lambda
        self.theta = None
        self.z = None
        self.u = None
        self.train_x = None
        self.sigma = 0.1

    def one_to_hot(self, y):
        self.classes = np.unique(y)
        Y = np.zeros([y.shape[0], len(self.classes)])
        for i, column in enumerate(self.classes):
            Y[np.where(y==column), i] = 1
        return Y

    def kernel(self, x, c):
        """kernel function"""
        return math.exp(-1*np.power(x-c, 2).sum()) / (2*self.h**2)

    def get_theta(self, train_y):
        """
        get specific theta
        :params train_y : the vector which describes whether the class is y or not (1 or 0).
        """
        return np.linalg.inv(self.K.T.dot(self.K) + self._lambda*np.eye(self.K.shape[0])).dot(self.K.T).dot(tr

    def fit(self, train_x, train_y):
        """cal each theta for one class"""
        self.train_x = np.array(train_x)
        train_y = np.array(train_y)
        train_y = train_y.reshape([len(train_y), 1])
        train_y = self.one_to_hot(train_y)

        # cal K
        self.K =  np.zeros(len(train_x)*len(train_x)).reshape([len(train_x), len(train_x)])

        for i in range(len(train_x)):
            for j in range(len(train_x)):
                self.K[i, j] = self.kernel(train_x[i], train_x[j])

        # cal theta for each class (theta is a matrix)

        theta_list = []
        for _class in range(train_y.shape[1]):
            theta = self.get_theta(train_y[:, _class])
            theta = theta.reshape((-1, 1))
            theta_list.append(theta)
        self.theta = np.concatenate(theta_list, axis=1)

    def predict(self, test_x):
        """
        predict funcation
        :return :predictions
        """
        # 基底をガウスカーネル基底に
```

```
        kernel_test_x = np.zeros([test_x.shape[0], self.train_x.shape[0]])

        for i in range(test_x.shape[0]):
            for j in range(self.train_x.shape[0]):
                kernel_test_x[i, j] = self.kernel(test_x[i], self.train_x[j])


        probabilities = np.matmul(kernel_test_x, self.theta)

        return np.argmax(probabilities, axis=1)
```

数字の画像データで検証

In [8]:

```
# train
train_list = []
for i in range(10):
    train_list.append(pd.read_csv("digit/digit_train%i.csv"% i, header =None))
train_target = []
for i in range(10):
    for j in range(500):
            train_target.append(i)
train = pd.concat(train_list)
train_x, train_y = shuffle(train, train_target)
train_x = np.array(train_x.reset_index().drop("index", axis = 1))

# test
test_list = []
for i in range(10):
    test_list.append(pd.read_csv("digit/digit_test%i.csv"% i, header =None))
test = pd.concat(test_list)
test_target = []
for i in range(10):
    for j in range(200):
            test_target.append(i)
test_x, test_y = shuffle(test, test_target)
test_x = np.array(test_x.reset_index().drop("index", axis = 1))
```

In [34]:

```
model = GaussKernelModel()
model.fit(train_x[:1000], train_y[:1000])
predict = model.predict(test_x[:500])
```

In [40]:

```
f1_score(test_y[:500], predict[:500], average="macro")
```

Out[40]:

0.93310818304631193

実験的に行なったがf-scoreが0.93と学習及び予測ができていることがわかる

homework2

$$B_\tau(y) = \sum_{y^{(\tau+1)}=1,,,y^{(m_i)}=1}^{c} exp(\sum_{k=\tau+2}^{m_i} \zeta^T \varphi(\mathbf{x_i}^k, y^k, y^{k-1}) + \zeta^T \varphi(\mathbf{x_i}^{\tau+1}, y^{\tau+1}, y))$$

は以下のように再帰表現できることを示せ

$$B_\tau(y^\tau) = \sum_{y^{(\tau+1)}=1}^{c} B_{\tau+1}(y^{\tau+1}) exp(\zeta^T \varphi(\mathbf{x_i}^{\tau+1}, y^{\tau+1}, y^\tau))$$

answer

proof):

$$\begin{aligned}
B_\tau(y^\tau) &= \sum_{y^{(\tau+1)}=1}^{c} B_{\tau+1}(y^{\tau+1}) exp(\zeta^T \varphi(\mathbf{x_i}^{\tau+1}, y^{\tau+1}, y^\tau)) \\
&= \sum_{y^{(\tau+1)}=1}^{c} \sum_{y^{(\tau+2)}=1}^{c} B_{\tau+2}(y^{\tau+2}) exp(\zeta^T \varphi(\mathbf{x_i}^{\tau+2}, y^{\tau+2}, y^{\tau+1})) exp(\zeta^T \varphi(\mathbf{x_i}^{\tau+1}, y^{\tau+1}, y^\tau))
\end{aligned}$$

ここで，一般に$e^a e^b = e^{a+b}$なので

$$\begin{aligned}
B_\tau(y^\tau) &= \sum_{y^{(\tau+1)}=1}^{c} \sum_{y^{(\tau+2)}=1}^{c} B_{\tau+2}(y^{\tau+2}) exp(\zeta^T \varphi(\mathbf{x_i}^{\tau+2}, y^{\tau+2}, y^{\tau+1}) + \zeta^T \varphi(\mathbf{x_i}^{\tau+1}, y^{\tau+1}, y^\tau)) \\
&= \sum_{y^{(\tau+1)}=1, y^{(\tau+2)}=1}^{c} B_{\tau+2}(y^{\tau+2}) exp(\zeta^T \varphi(\mathbf{x_i}^{\tau+2}, y^{\tau+2}, y^{\tau+1}) + \zeta^T \varphi(\mathbf{x_i}^{\tau+1}, y^{\tau+1}, y^\tau))
\end{aligned}$$

以上の操作を繰り返すと,

$$B_{m_i}(y^{m_i}) = \sum_{y^{(\tau+1)}=1}^{c} exp(\zeta^T \varphi(\mathbf{x_i}^{m_i}, y^{m_i}, y^{m_i-1}))$$

であるから,

$$B_\tau(y) = \sum_{y^{(\tau+1)}=1,,,y^{(m_i)}=1}^{c} exp(\sum_{k=\tau+2}^{m_i} \zeta^T \varphi(\mathbf{x_i}^k, y^k, y^{k-1}) + \zeta^T \varphi(\mathbf{x_i}^{\tau+1}, y^{\tau+1}, y))$$

よって，題意は満たされた。

【Q.E.D】

homework3

$$P_\tau(y) = \max_{y^1,,,y^{\tau-1}\in(1,2,,,c)} \left[\sum_{k=1}^{\tau-1} \zeta^T \varphi(\mathbf{x_i}^k, y^k, y^{k-1}) + \zeta^T \varphi(\mathbf{x_i}^\tau, y, y^{\tau-1}) \right]$$

は以下のように再帰表現できることを示せ

$$P_\tau(y^\tau) = \max_{y^{\tau-1}\in(1,2,,,c)} \left[P_\tau(y^{\tau-1}) + \zeta^T \varphi(\mathbf{x_i}^\tau, y^\tau, y^{\tau-1}) \right]$$

answer

proof):

$$\begin{aligned} P_\tau(y^\tau) &= \max_{y^{\tau-1}\in(1,2,,,c)} \left[P_\tau(y^{\tau-1}) + \zeta^T \varphi(\mathbf{x_i}^\tau, y^\tau, y^{\tau-1}) \right] \\ &= \max_{y^{\tau-1}\in(1,2,,,c)} \left[\max_{y^{\tau-2}\in(1,2,,,c)} \left[P_{\tau-1}(y^{\tau-2}) + \zeta^T \varphi(\mathbf{x_i}^{\tau-1}, y^{\tau-1}, y^{\tau-2}) \right] + \zeta^T \varphi(\mathbf{x_i}^\tau, y^\tau, y^{\tau-1}) \right] \end{aligned}$$

ここで,

$$\max_{y^{\tau-2}\in(1,2,,,c)} \left[P_{\tau-1}(y^{\tau-2}) + \zeta^T \varphi(\mathbf{x_i}^{\tau-1}, y^{\tau-1}, y^{\tau-2}) \right]$$

の値は, $\zeta^T \varphi(\mathbf{x_i}^\tau, y^\tau, y^{\tau-1})$ によらないので,

$$\max_{y^{\tau-2}\in(1,2,,,c)} \left[P_{\tau-1}(y^{\tau-2}) + \zeta^T \varphi(\mathbf{x_i}^{\tau-1}, y^{\tau-1}, y^{\tau-2}) \right] = \max_{y^{\tau-2}\in(1,2,,,c)} \left[P_{\tau-1}(y^{\tau-2}) + \zeta^T \varphi(\mathbf{x_i}^{\tau-1}, y^{\tau-1}, y^{\tau-2}) + \zeta^T \varphi(\mathbf{x}$$

よって,

$$\begin{aligned} P_\tau(y^\tau) &= \max_{y^{\tau-1}\in(1,2,,,c)} \left[P_\tau(y^{\tau-1}) + \zeta^T \varphi(\mathbf{x_i}^\tau, y^\tau, y^{\tau-1}) \right] \\ &= \max_{y^{\tau-2},y^{\tau-1}\in(1,2,,,c)} \left[P_{\tau-1}(y^{\tau-2}) + \zeta^T \varphi(\mathbf{x_i}^{\tau-1}, y^{\tau-1}, y^{\tau-2}) + \zeta^T \varphi(\mathbf{x_i}^\tau, y^\tau, y^{\tau-1}) \right] \end{aligned}$$

以上の操作を繰り返すと, homework2と同じように,

$$P_\tau(y) = \max_{y^1,,,y^{\tau-1}\in(1,2,,,c)} \left[\sum_{k=1}^{\tau-1} \zeta^T \varphi(\mathbf{x_i}^k, y^k, y^{k-1}) + \zeta^T \varphi(\mathbf{x_i}^\tau, y, y^{\tau-1}) \right]$$

よって, 題意は示された。

【Q.E.D】

In [ ]: